MEMORIA DINÁMICA

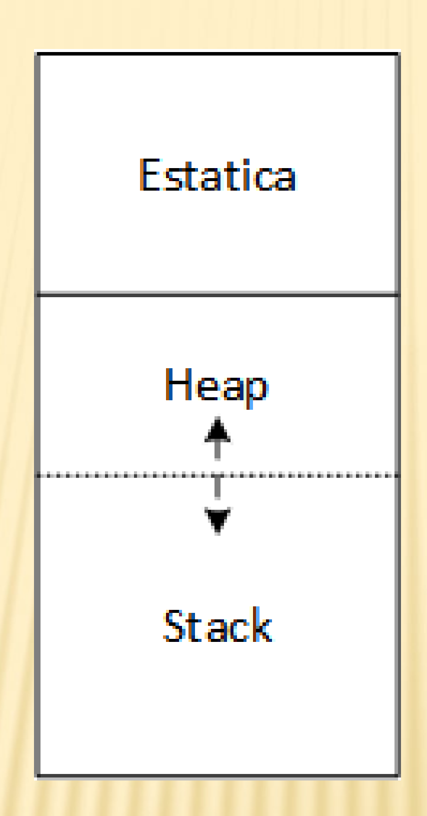
En muchas aplicaciones no se conoce con anticipación la cantidad de memoria necesaria para las variables a utilizar en un programa.

No se conoce los datos que se procesarán

Ubicarlos en diversas posiciones de la memoria para ejecutarlo mas velozmente.

ASIGNACIÓN DINÁMICA DE MEMORIA

El compilador divide a la memoria en tres zonas



Estática: donde se ubica el código, las variables globales.

Stack (pila): variables locales y direcciones de retorno de funciones.

eap (zona de memoria libre)

FUNCIÓN MALLOC()

Asignar memoria libre para la cantidad de espacio requerido.

Prototipo

***malloc(unsigned int tamaño_requerido);**

naño_requerido: cantidad de bytes solicitados

Devuelve un puntero a void que contiene la dirección de memoria en donde ésta ubicado el bloque asignado por ella. En caso de que no exista lugar disponible en la memoria para asignar dicha solicitud retorna un puntero nulo.

Un puntero a void es un puntero que no apunta a ningún tipo específico, éste se incrementa en un byte. Normalmente se debe realizar un casting de dicho puntero.

FUNCIÓN FREE()

Libera un bloque de memoria pedido por la función malloc().

Prototipo

d free(void *bloque_pedido);

bloque_pedido: dirección del bloque de memoria solicitado.

SIZEOF

Devuelve el tamaño en bytes de la expresión o del tipo dado.

Sintaxis

zeof <expresión>

O

zeof (<tipo>)